Worst Case Modeling of Aggregate Scheduling by Network Calculus

Ulrich Klehmet Rüdiger Berndt
Computer Networks and Communication Systems
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
Email: {ulrich.klehmet, ruediger.berndt}@fau.de

***Abstract*—*Network Calculus* (NC) is a powerful mathematical theory for the performance evaluation of communication systems, since it allows to obtain worst-case performance measures. In communications system modeling, the NC theory is often used to determine *Quality of Service* (QoS) guarantees, for example of packet-switched systems. In networking systems, the aggregation of data flows plays an important role while modeling the multiplexing scheme. When the multiplexing order is not *First in, First out* (FIFO), the strictness of the service curve plays an important role. This article deals with problems that arise from the strictness requirement considering aggregate scheduling. The literature reports that the strictness of an aggregated service curve is a fundamental precondition to obtain the individual service curve for a single left-over flow when a node processes multiple input flows in a non-FIFO manner. In many publications, this important strictness property is assumed to be a feature of the service curve only. We will show that, in general, this assumption is not true. In most cases, only the concrete input flow in connection with the service curve allows to decide whether the service curve is strict or non-strict. However, the abstraction from a concrete input flow with an arrival curve as upper bound is not enough to determine the service curve's strictness. Therefore, to bypass the strict-non-strict problems, we devise theorems to gain guaranteed performance values for a left-over flow.**

Keywords– Worst-case Communication System Modeling; Network Calculus; Aggregate Scheduling; Strict-non-strict Service Curves; Backlogged Period

I. INTRODUCTION

For systems with hard real-time requirements, *timeliness* plays an important role. This *Quality of Service* (QoS) requirement can be found in various kinds of systems, including cars, airplanes, industrial networks, or power plants [1]. Analytical performance evaluation of such systems cannot be based on stochastic modeling, like traditional queuing theory, only. Since worst-case performance parameters like maximum delay of service times [2] are required, the knowledge of mean values is not sufficient. In other words, one needs a mathematical tool to calculate performance figures—in terms of bounding values—which are valid in any case. Such a tool is *Network Calculus* (NC), as a novel system theory for deterministic queuing systems [3] [4].

This article is structured as follows: In Section II, we describe the basic elements of NC. Section III introduces aggregate scheduling and the use of NC w.r.t. the analysis of individual flows as part of an aggregation. Section IV shows possibilities to overcome the problems described in Section III. Finally, in Section V we draw a conclusion.

II. BASIC MODELING ELEMENTS OF NETWORK CALCULUS

The most important modeling elements of NC are given by *arrival curves*, *service curves*, and the *min-plus convolution*. Arrival and service curves are the basis for the computation of maximum deterministic boundary values, like backlog bounds, or delay bounds [4].

Definition 1 (Arrival curve): Let $\alpha(t)$ be a non-negative, non-decreasing function. Flow F with input $x(t)$ at time t is constrained by the arrival curve $\alpha(t)$ iff:

$$\forall\ 0 \leq s \leq t : x(t) - x(s) \leq \alpha(t-s) \tag{1}$$

Flow F is also called α-smooth.

Example 1: A commonly used arrival curve is the token bucket constraint:

$$\alpha_{r,b}(t) = \begin{cases} r \cdot t + b & \text{for } t > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Figure 1 shows a token bucket arrival curve (dashed blue line) forming an upper limit for a traffic flow $x(t)$ with an average rate r and an instantaneous burst b. This means for $\Delta t := t-s$ and $\Delta t \to 0$, $\lim_{t\to s}\{x(t)-x(s)\} \leq \lim_{\Delta t\to 0}\{r\cdot\Delta t+b\} = b$.

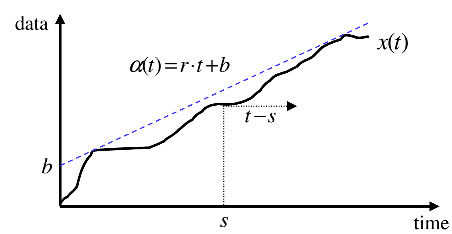

Figure 1. Token bucket arrival curve

Next to arrival curves, the convolution operation plays an important role in NC theory.

Definition 2 (Min-Plus Convolution): Let $f(t)$ and $g(t)$ be non-negative, non-decreasing functions that are 0 for $t \leq 0$. A third function, called *min-plus convolution* is defined as:

$$(f \otimes g)(t) = \inf_{0\leq s\leq t}\{f(s) + g(t-s)\} \tag{3}$$

By applying this operation, arrival curve $\alpha(t)$ can be characterized with respect to $x(t)$ as:

$$x(t) \leq (x \otimes \alpha)(t) \tag{4}$$

Service curves, in contrast to arrival curves, are used to model the output of a system—for example to determine whether there is a guaranteed minimum output $y(t)$.

 ISBN: 978-1-61208-398-8

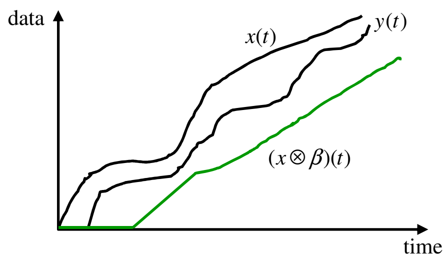

Figure 2. Convolution as a lower output bound

Definition 3 (Service Curve): Let system S with input flow $x(t)$ and output flow $y(t)$ be given. The system has a (minimum) service curve $\beta(t)$, iff $\beta(t)$ is a non-negative, non-decreasing function with $\beta(0) = 0$ and

$$y(t) \geq (x \otimes \beta)(t) \quad (5)$$

Figure 2 shows $(x \otimes \beta)(t) = \inf_{0 \leq s \leq t}\{x(s) + \beta(t-s)\}$ as lower bound of output $y(t)$ w.r.t. input $x(t)$.

Example 2: The commonly used *rate-latency function* reflects a service element which offers a minimum service of rate R after a worst-case latency of T, by doing so, all internal behavior of the service is hidden and only the worst-case is described.

$$\beta(t) = \beta_{R,T}(t) = R \cdot [t-T]^+ := R \cdot \max\{0; t-T\} \quad (6)$$

In Figure 4, the graph of $\beta_{R,T}(t)$ (green) depicts the rate-latency service curve with rate R and latency T.

Consider a system S with input flow $x(t)$, arrival curve $\alpha(t)$, output flow $y(t)$ and service curve $\beta(t)$. Then, according to [4], the following three bounds can be calculated.

Proposition 1 (Backlog, Delay and Output bound):

- Backlog bound v:
  $v(t) = x(t) - y(t) \leq \sup_{s \geq 0}\{\alpha(s) - \beta(s)\}$
- Delay bound $d(t)$ of input x (FIFO service):
  $d \leq \sup_{s \geq 0}\{\inf\{\tau : \alpha(s) \leq \beta(s+\tau)\}\}$
- Output bound $\alpha^*(t)$:
  $\alpha^*(t) = \alpha \oslash \beta := \sup_{s \geq 0}\{\alpha(t+s) - \beta(s)\}$

Figure 3 depicts d and v for a general arrival- and service curve.

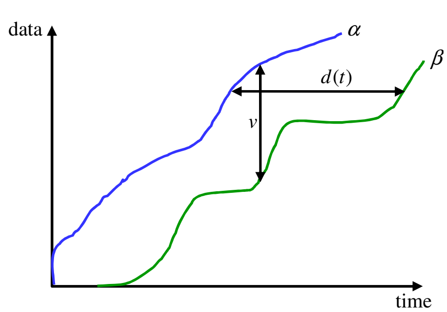

Figure 3. Backlog and delay bound

Example 3: Suppose a system with token bucket input and rate-latency service, according to (1), (3), (5), and based on

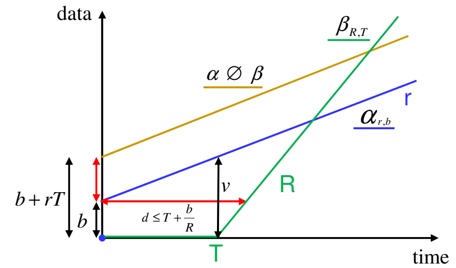

Figure 4. Example for all bounds, cf. Proposition 1

Proposition 1 the delay bound is given by $d \leq \frac{b}{R} + T$, the output bound by $\alpha^*(t) = r(t+T) + b$, and the backlog is bounded by $v = b + rT$ (see Figure 4).

III. Aggregate Scheduling

Up to now, only single flow-based scheduling has been considered. But, in many real-world systems *aggregate scheduling* is used [5]. We speak of aggregate scheduling whenever at least two streams are handled as a single stream. While in [6], delay bounds for FIFO networks are given, in [7] the main goal is to derive end-to-end delay bounds for general multiplexing. Important applications of aggregate scheduling are, among others, *differentiated service* domains (DS) of the Internet, the determination of safety-critical delay bounds within automotive intra car-communication [8], and diverse time-critical industrial applications [9].

In order to apply NC to such communication networks, we have to expand the rules to multiplexing and aggregate scheduling.

Assume that n flows enter a system (network) or system node which are scheduled by aggregation. According to [10] the aggregate input flow and arrival curve are given as follows.

Definition 4 (Aggregation, Multiplexing): An *aggregation* (also: *multiplexing*) of n flows can be expressed by adding the single input flows and arrival curves, respectively: let $n = 2$, then the aggregated input flow is $x(t) = x_1(t) + x_2(t)$ and $\alpha(t) = \alpha_1(t) + \alpha_2(t)$, where x_1, x_2 and α_1, α_2 are the corresponding single input flows and arrival curves.

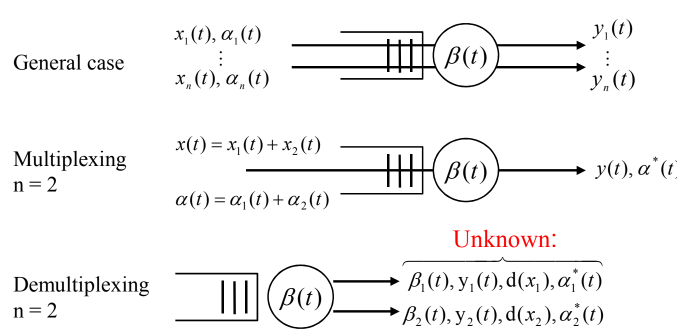

Figure 5. Multiplexing of flows: input x_i, output y_i, arrival & service curve α_i, $\beta(t) = \beta_{aggr}$

Considering multiplexed streams (Figure 5) the question is whether Proposition 1 might be applied to the individual

 ISBN: 978-1-61208-398-8

streams of an aggregation, for example to calculate the maximum delay of a single flow x_i.

Firstly, this depends on the type of aggregate scheduling, like FIFO (see [11]), priority-scheduling, or unknown arbitration, and secondly on the service curve β_{aggr} of the aggregated flow. If no knowledge about the choice of service between the flows is present, then we speak of *arbitrary multiplexing* [12] (also: *blind multiplexing*), and the situation is more complex. In such cases, the distinction between *strict* and *non-strict* aggregate service curves plays an important role [4].

Proposition 2 (Blind Multiplexing): Let a node be serving the flows x_1 and x_2 with an unknown arbitration. Assume that the node guarantees a strict service curve β to the aggregation of the two flows, and that flow x_2 is bounded by α_2. Let $\beta_1(t) := [\beta(t) - \alpha_2(t)]^+$; β_1 is a service curve for flow x_1 if it is wide-sense increasing.

Definition 5 (Strictness of service curve): A system S offers a *strict* service curve β to a flow, if during any backlogged period $[s, t]$ of duration $t - s$ the output y of the flow is at least equal to $\beta(u)$, i.e., $y(t) - y(s) \geq \beta(t - s)$. Obviously, any strict service curve is also a service curve.

Example 4: Figure 6 (left) shows a token bucket input $x = rt + b$ and a service curve $\beta(t) = R \cdot t$. Here, the output $y(u) \geq \beta(u)$ in all backlogged periods u: $u \leq$ busy period. Thus, β is *strict* (there is only one backlogged period u).

If we switch to the rate-latency service curve $\beta_{R,T}(t) = R \cdot [t - T]^+$ (Figure 6, right), we get a *non-strict* service curve. The backlogged period starts at 0 and never ends: because in the worst case, all input data of x remains in the system for time T before being served with rate R, but new data of x always arrives during T. The definition of the service curve specifies the output y as $y(t) \geq (x \otimes \beta)(t)$. Indeed, it is valid that the output $y(u_0) \geq \beta_{R,T}(u_0)$, but this is not guaranteed regarding the backlogged period $u > u_0$. Thus, it is possible that $y(u) \ngeq \beta_{R,T}(u)$ as $(x \otimes \beta_{R,T})(u) - (x \otimes \beta_{R,T})(0) = (x \otimes \beta_{R,T})(u) < \beta_{R,T}(u) = \beta_{R,T}(u) - \beta_{R,T}(0)$ if $T > 0$. In this scenario, β is *non-strict*.

The above example raises the issue whether there are classes of service functions that are always *strict* or *non-strict*, respectively.

In literature, the service curve $\beta_{R,T}(t) = R \cdot [t - T]^+$ (or even any *convex* service curve) is often used as a strict service curve per se; see, for instance, [13]. We will see that both, *strictness* and *non-strictness*, are not only based on the service curve itself but also on the corresponding input flow x. Considering aggregate flow situations, this means that the strictness of an aggregated input flow needs to be checked before applying the important Proposition 2. Therefore, the condition $y(u) \geq \beta(u)$, $\forall$ backlogged periods u needs to be proofed.

First, we will provide and prove some characterizations for applications using token bucket input flows and the commonly used rate-latency service curves.

Theorem 1 (Non-strict functions): Let a system with rate-latency service curve $\beta_{R,T}$ and token bucket arrival curve $\alpha_{r,b}$ with $r < R$ and $T > 0$ be given. Furthermore, the worst case scenario is assumed: the input is served with minimum rate R after a possible maximum delay T. We claim that the service curve $\beta_{R,T}$ cannot be strict, if the input flow $x(t)$ is a *strictly increasing* function.

Proof:
Assume $\beta_{R,T}$ is strict. Based on Proposition 1 we know $\alpha^*(t) = \alpha \oslash \beta := \sup_{s \geq 0}\{\alpha(t+s) - \beta(s)\}$, here $\alpha^*(t) = r(t+T) + b$. Because $r < R$, there is a point in time t_s, so that $\beta_{R,T}(t_s) = \alpha^*(t_s)$ and $\beta_{R,T}(t) > \alpha^*(t)$ if $t > t_s$, i.e., $\forall\ t_0 > t_s : \beta_{R,T}(t_0) - \alpha^*(t_s) \geq \alpha^*(t_0) - \alpha^*(t_s)$ $\Rightarrow \Delta\beta_{R,T} = \beta_{R,T}(t_0) - \beta_{R,T}(t_s) \geq \alpha^*(t_0) - \alpha^*(t_s) = \Delta\alpha^*$. Since $x(t)$ is strictly increasing and latency $T > 0$: for any $t_0 > t_s$: $u := t_0 - t_s$ is a backlogged period—cf. Figure 6 (right).
$\beta_{R,T}$ is supposed to be strict, so output $y(u) \geq \beta_{R,T}(u) = \beta_{R,T}(t_0) - \beta_{R,T}(t_s) \geq \alpha^*(t_0) - \alpha^*(t_s) = \alpha^*(t_0 - t_s)$. But this is a contradiction to α^* being an arrival curve for output y. Thus, the assumption is wrong, i.e., $\beta_{R,T}$ is non-strict. $\square$

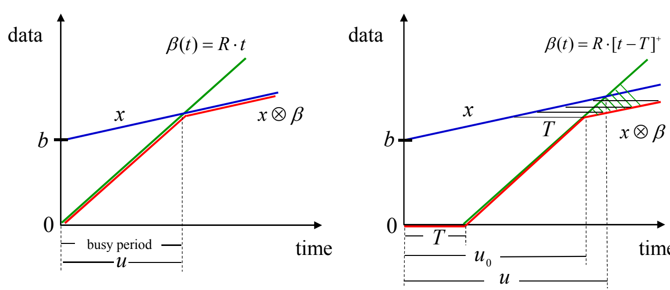

Figure 6. Strict and non-strict service curve

Unfortunately, the feature of being a non-strictly increasing input x is not a sufficient condition for a strict service curve $\beta_{R,T}$: using the same token bucket arrival curve $\alpha_{r,b}$ and rate-latency service curve $\beta_{R,T}$, one can find non-strictly increasing input functions x that make the service curve $\beta_{R,T}$ either strict or non-strict. This will be demonstrated by the following example.

Example 5: Let

$$\alpha_{r,b} := \begin{cases} 1,5t + 5 & \text{for } t > 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

and $\beta_{R,T} := 2(t-2)^+$. Furthermore, let the input x be first identical to $\alpha_{r,b}$, and then stagnate at time t'. The parameter t' is computed using equation $\alpha_{r,b}(t) = \beta_{R,T}(t + T)$. This guarantees that no displacement of the $\beta_{R,T}$ graph within the convolution graph of $x \otimes \beta_{R,T}$ occurs:
$1,5t + 5 = 2((t+2) - 2); t = t' = 10$ fulfills this equation. So, we define the input as

$$x(t) = \begin{cases} 0 & \text{for } t \leq 0 \\ 1,5t + 5 & \text{for } t \leq 10 \\ 20 & \text{otherwise} \end{cases} \tag{8}$$

Result: The service curve $\beta_{R,T}$ is strict (Figure 7).

 ISBN: 978-1-61208-398-8

Now, the input x is changed from x to $\tilde{x}$ ($\tilde{x}$ is still non-strictly increasing):

$$\tilde{x}(t) = \begin{cases} 0 & \text{for } t \leq 0 \\ 0,75t + 2,5 & \text{for } t \leq 10 \\ 10 & \text{otherwise} \end{cases} \quad (9)$$

Result: The same service curve $\beta_{R,T}$ is now non-strict (Figure 7).

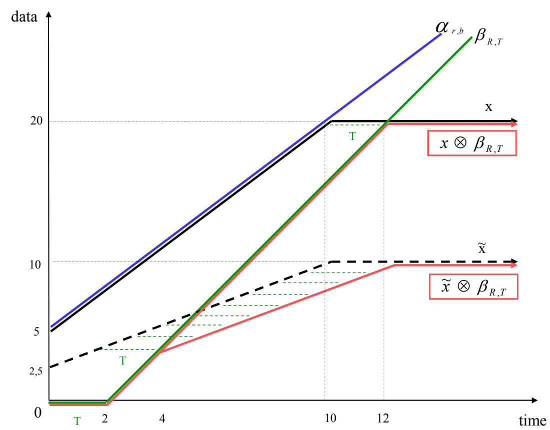

Figure 7. Input x changed to $\tilde{x}$ causes non-strictness

Obviously, all **input functions** $\boldsymbol{x}$ of the form (or multiple repetitions)

$$x(t) = \begin{cases} mt + n & \text{for } t \leq t' \\ const & \text{otherwise} \end{cases} \quad (10)$$

cause the service curve $\beta_{R,T}$ to be strict, if the constant part of x starts within or on the border of the red-dashed triangle of Figure 8.

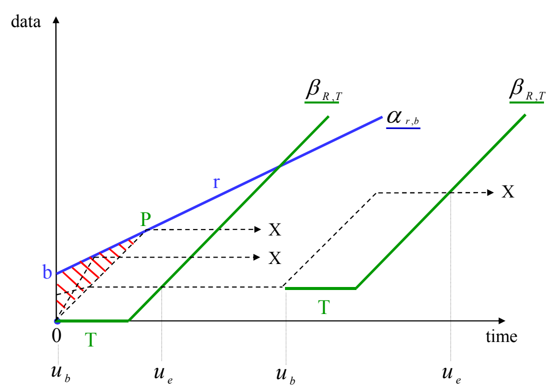

Figure 8. Input area making $\beta_{R,T}$ strict

Here, u_b is the begin and u_e the end of a backlogged period, b is the burst size of the arrival curve $\alpha_{r,b}$ and $P = P(\acute{t}, \acute{y})$ with $\acute{t}$: $\alpha_{r,b}(\acute{t}) = \beta_{R,T}(\acute{t} + T)$, i.e., the intersection of $\alpha_{r,b}$ with the parallel line to $\beta_{R,T}$, given by the curve $y = Rt$.

So, we state: the feature of being *strict* or *non-strict* is not only determined by a service curve itself. In all cases, this feature depends on both, the service curve *and* the actual input x. Since often concrete input flows remain unknown, this aggravates NC's application to worst-case analysis of scenarios including aggregate scheduling.

IV. Bypassing the Strict-Non-Strict Problem

In the following, we try to overcome the problem of strictness or non-strictness in case of aggregate scheduling. Our aim is to look for and—if possible—create a service curve for a single flow x_i of an aggregate flow x without any consideration, either on FIFO or non-FIFO scheduling, and strictness (non-strictness) of the aggregate-service curve $\beta_{aggr}(t)$.

All together, these considerations provoke the following statement that determines a generalization of Proposition 2 as it does not differentiate between strict and non-strict:

Theorem 2 (Construction of always-to-use service curves): Consider a node with some unknown arbitration serving the flows x_1 and x_2. Let x be the aggregated input with $x = x_1 + x_2$ and $y = y_1 + y_2$ the aggregated output, respectively. Assume that the node offers a service curve β to the aggregation of the two flows with $y(t) \geq (x \otimes \beta)(t)$, and flow x_2 is α_2-smooth. Define $\beta_1(t) := [(x \otimes \beta)(t) - \alpha_2(t)]^+$. If β_1 is wide-sense increasing, then β_1 is a service curve for flow x_1.

Proof:
Since $\beta = \beta_{aggr}$ is a service curve for input x, we know:
(i) $y \geq x \otimes \beta_{aggr}$
Let $\Phi := \{f \mid f(t_1) \geq f(t_0) \text{ for } t_1 \geq t_0,\ f(t) = 0 \text{ for } t < 0,\ f(t) \text{ left-continuous},\ t \in \mathbb{R}\}$
In [4] we find the following algebraic rule:
If $f(0) = g(0) = 0$ then $(f \otimes g \leq min\{f, g\} \Rightarrow$
$x \otimes [(x \otimes \beta_{aggr})] \leq min\{x, (x \otimes \beta_{aggr})\} \Rightarrow$
$x \otimes [(x \otimes \beta_{aggr})] \leq x \otimes \beta_{aggr} \leq y$.
Thus, $x \otimes [(x \otimes \beta_{aggr})] \leq y \Rightarrow$ expression $x \otimes \beta_{aggr}$ itself is a service curve for x, and together with (i) it is even *strict*. But this means: $\beta_1(t) := [(x \otimes \beta)(t) - \alpha_2(t)]^+$ is a service curve for flow x_1 if β_1 is wide-sense increasing. □

Based on Theorem 2, we can derive the construction of a strict service curve.

Theorem 3 (Construction of a strict service): Suppose a node with the conditions of Theorem 2 and with a service curve β to the aggregate with $y(t) = (x \otimes \beta)(t)$ instead of $y(t) \geq (x \otimes \beta)(t)$.

Then the following is always valid:
(i) $\beta^* := (x \otimes \beta)$ is a strict service curve to the flow x
(ii) $\beta^* := (x \otimes \beta)$ is the greatest strict service curve to x

Proof:
(i): β is service curve, i.e., $y \geq x \otimes \beta \Rightarrow y(t) \geq \beta^*(t)$ which means β^* is strict.
(ii): Assume β^* is not the greatest service curve, that means there is a β' with $\beta \geq \beta' > \beta^*$ and β' is strict. Then $y \geq \beta'$, and $\Rightarrow$ $y = x \otimes \beta \geq \beta'$. Since $\beta^* := (x \otimes \beta) \Rightarrow \beta^* \geq \beta'$ which contradicts the assumption. □

Thus, we can state the following: the construction of a strict service curve is possible in case of blind multiplexing with non-strict service curve to the aggregate input. Of course, if $y(t) \geq (x \otimes \beta)(t)$ and not $y(t) = (x \otimes \beta)(t)$, then it may be that a greater strict service curve than $\beta^* := (x \otimes \beta)$ exists

 ISBN: 978-1-61208-398-8

(and therefore a better one w.r.t. β_i of a single flow x_i); but in any case, it is a strict service curve.

Now we give another theorem to overcome the question of strictness or non-strictness of a service curve. Due to Theorem 1, the important service curve $\beta_{R,T}$ cannot be strict, if input flow $x(t)$ is a strictly increasing function. And, according to Figure 7, the change of input x to $\tilde{x}$ causes the service curve $\beta_{R,T}$ to be non-strict although $\beta_{R,T}$ is not changed. Consequently, the assumption of Proposition 2 would not be fulfilled—but sometimes it is possible to characterize an (aggregated) input $x = x_1 + x_2$ so that a service curve $\beta_1(t)$ of flow x_1 exists, even if the service curve β of the aggregate is non-strict:

Theorem 4 (Singular flow bounded by K): Consider a node serving the flows x_1 and x_2 with some unknown arbitration between the two flows. Let x be the aggregated input with $x = x_1 + x_2$ and $y = y_1 + y_2$ the aggregated output, respectively. Assume that the node offers a service curve β to the aggregate of the two flows, and let flow x_2 be bounded by $K > 0$. Define $\beta_1(t) := [\beta(t)] - K]^+$. If β_1 is wide-sense increasing, then β_1 is a service curve for flow x_1.

Proof:
Let $\Phi := \{f \mid f(t_1) \geq f(t_0) \text{ for } t_1 \geq t_0,\ f(t) = 0 \text{ for } t < 0,\ f(t) \text{ left-continuous},\ t \in \mathbb{R}\}$
According to the algebraic rules in [4]:
Rule a): $K + (g \otimes f) = g \otimes (K + f)$ if $g, f \in \Phi,\ K \in \mathbb{R}^+$
Rule b): If $f \leq f^{'}$ and $g \leq g^{'} \Rightarrow f \otimes g \leq f^{'} \otimes g^{'}$ for $g, g^{'}, f, f^{'} \in \Phi$, β is service curve, i.e., $[y_1(t) + y_2(t)] \geq ((x_1 + x_2) \otimes \beta)(t)$
$\Rightarrow y_1(t) \geq ((x_1+x_2) \otimes \beta)(t) - y_2(t) \Rightarrow$ Because $x_1 + x_2 \geq x_1$ and Rule b):
$\Rightarrow y_1(t) \geq (x_1 \otimes \beta)(t) - y_2(t)$. Always is true $y_2(t) \leq x_2(t)\ \forall t \Rightarrow y_1(t) \geq (x_1 \otimes \beta)(t) - x_2(t)$, and since by assumption $x_2(t) \leq K \Rightarrow y_1(t) \geq (x_1 \otimes \beta)(t) - K$.
Applying Rule a) we get $(x_1 \otimes \beta)(t) - K = (x_1 \otimes (\beta - K)(t)$
$\Rightarrow y_1(t) \geq (x_1 \otimes (\beta - K)(t)$; which means $\beta_1 := [\beta - K]^+$ is service curve of x_1 if it is wide-sense increasing.□

Remark:
Now, we can state that all (aggregate-) input functions $x = x_1 + x_2$ with $x_2(t) \leq K$ possess the service curve $\beta_1 := [\beta - K]^+$ independent from strictness or non-strictness of β, as for example of the aggregate x:

$$x := \begin{cases} mt + n & \text{for any } t \in \mathbb{R}^+ \\ const & \text{otherwise} \end{cases} \tag{11}$$

Example 6: A typical scenario of blind multiplexing is given by two input flows x_{low} and x_{high} with a worst case service situation for flow x_{low}—also called *preemptive priority schedule*, i.e., x_{high} will be served first, and will always interrupt the x_{low}-service as soon as data of flow x_{high} is present. Only if there is no data of flow x_{high}, flow x_{low} will be served.

Proposition 2, Theorem 2, and Theorem 4 may provide a service curve β_{low} for the low-priority flow x_{low}. An important question is whether this (minimum) service curve β_{low}

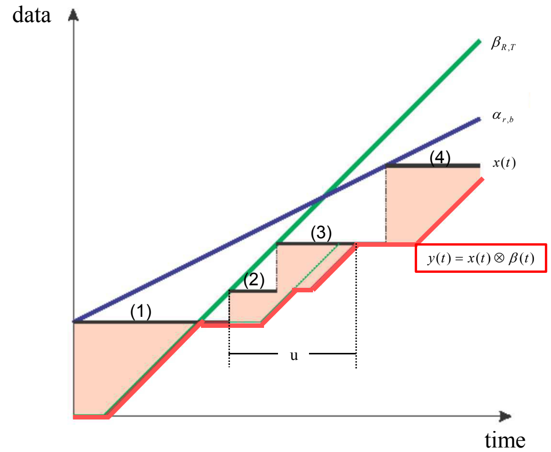

Figure 9. Finite backlogged period u—but $\beta_{R,T}$ is not strict

together with α_{low}— if known—can be used to compute the maximal delay, the maximal backlog bound, or the maximal output bound of the flow x_{low} by applying Proposition 1. In terms of backlog bound we can argue that the service facility is work-conserving, i.e., the 'unfinished work' $x(t) - y(t)$ depends only on the arrival instants and the data(packet-)length, and not on the order of service. Therefore, it is easy to realize that Proposition 1 (Backlog bound) is applicable here:

$$x_{low}(t) - y_{low}(t) \leq \sup_{s \geq 0}\{\alpha_{low}(s) - \beta_{low}(s)\} \tag{12}$$

Nevertheless, relating to x_{low}-traffic, both of the other bounds, i.e., Delay and Output bound—are applicable, too. For example, considering the delay bound of x_{low}:

$$d \leq \sup_{t \geq 0}\{\inf\{\tau : \alpha_{low}(t) \leq \beta_{low}(t + \tau)\}\} \tag{13}$$

Here, the *virtual delay*

$$d_\tau(t) = \inf\{\tau \geq 0 : \alpha_{low}(t) \leq \beta_{low}(t + \tau)\} \tag{14}$$

ensures that an input x which arrives at time t will leave service not later than $d_\tau(t)$. This is guaranteed for FIFO scheduling but not for blind multiplexing. However, we may presume FIFO per single flow (e.g. x_{low} within the aggregate of blind multiplexing) and thus apply all three bounding theorems without any restrictions.

The important statement $d \leq \sup_{s \geq 0}\{\inf\{\tau : \alpha(s) \leq \beta(s + \tau)\}\}$ is only valid for FIFO systems. Hence the question is whether it is possible to state a similar proposition for systems in general—either FIFO or not. The next theorem is a first answer to this.

Theorem 5 (Delay bounds in general): Let a system S with input x, arrival curve α, output y, service curve β, and point in time t_α with $\forall t > t_\alpha : \alpha(t) < \beta(t)$ be given. Furthermore, let $U = \{u | u \text{ is backlogged period}\}$, and $l(u)$ be the length of a backlogged period. If the service curve β is strict or u is finite $\forall u \in U$, then the maximal delay d is given by

$$d \leq \sup_{u \in U}\{l(u) : (x \otimes \beta)(t) \leq x(t) \wedge t \in u\} \tag{15}$$

 ISBN: 978-1-61208-398-8

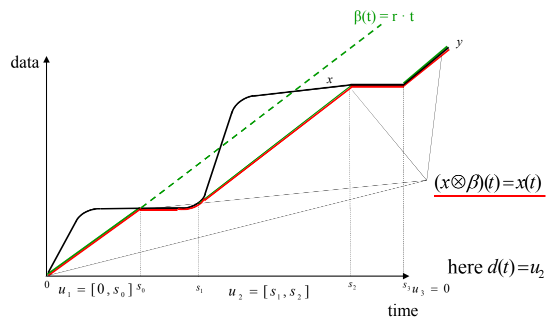

Figure 10. Strict service curve

Proof:
Case: u is finite $\forall u \in U$—proof is trivial.
Case: service curve β is strict: due to the subsequent Lemma the backlogged period is finite. And otherwise a backlogged period is finite at time t if $x(t) = y(t)$, cf. Figure 10. Take any backlogged period u and let s be start and t be the end of the backlogged period $u = [s, t]$, that means for input x and output y: $y(s) = x(s)$ and $y(t) = x(t)$. Let β be the service curve, i.e., output y is lower-bounded by $x \otimes \beta$; now, $\forall$ subsets $v \subseteq u$ with $v := [s, z]$ and $z \leq t$:
(i) $x(z) \geq y(z) \geq (x \otimes \beta)(z)$.
On the other hand: $\forall\ \Delta_v := z - s$, and due to strictness of β: $y(\Delta_v) \geq \beta(\Delta_v) = \beta(z-s)$. That means we get the following equation:
(ii) $\inf_{s \leq \tilde{t} \leq z}\{x(s) + \beta(\tilde{t} - s)\} = x(s) + \beta(\tilde{t} - s) = (x \otimes \beta)(\tilde{t})$.
This is because at the points from s to z inside the finite backlogged period $u = [s, t]$, the value of convolution $x \otimes \beta$ is determined by the service curve β alone. Therefore, for $z \to t$ we get:
(iii) $y(z) = y(t) = (x \otimes \beta)(t)$ and $y(t) = x(t) \ \Rightarrow x(t) = (x \otimes \beta)(t)$ where t is the end of backlogged period u.
$\Rightarrow l(v) \leq l(u)$ with $(x \otimes \beta)(z) \leq x(z)$.
$\Rightarrow$ Maximal delay d: $d \leq \sup_{u \in U}\{l(u) : (x \otimes \beta)(t) \leq x(t) \wedge t \in u\}$. $\square$

Lemma: Let a system *S* with input x, arrival curve α, output y, service curve β, and point in time t_α with $\forall t > t_\alpha$: $\alpha(t) < \beta(t)$ be given. If the service curve β is strict, then any backlogged period is finite.

Proof:
Since β is strict we have $y(t) \geq \beta(t)\ \forall t$.
Of course $y(t) \leq x(t)$ is always true, altogether

$$(x \otimes \beta)(t) \leq \beta(t) \leq y(t) \leq x(t) \quad \forall t \tag{16}$$

Let u be any backlogged period, and suppose u is *not* finite. Then $y(t) < x(t)\ \forall t$ and, therefore, also $\forall t > t_\alpha$.
From (16): $(x \otimes \beta)(t) \leq \beta(t) \leq y(t) < x(t) \leq \alpha(t - 0) = \alpha(t)$. But that means $\beta(t) < \alpha(t)\ \forall t > t_\alpha$ which contradicts the precondition of the Lemma. Thus, u is finite, i.e., $\exists\ t_0$: $y(t_0) = x(t_0)$. $\square$

Unfortunately, the opposite statement is not valid as shown in Figure 9.

V. Conclusions

This paper deals with worst case modeling of aggregate scheduling. We want to get guaranteed performance parameters, like maximal end-to-end delay of individual so-called left-over flows of an aggregate. When using the analytical tool Network Calculus (NC), among others the service curve is required as main modeling element. In case of blind multiplexing the following particular problem occurs: the construction of a service curve for the single output after demultiplexing an aggregated flow $x = x_1 + x_2$ requires the *strictness* of the aggregated service curve.

In publications like [13] [12], or others, it is assumed that the rate latency service curve $\beta_{R,T}$ (very often used as aggregated service curve) fulfills the strictness property.

In this article, we demonstrated that the property of being *strict* or *non-strict* does not depend on the service curve solely. Only in combination with the concrete input—or at least with a special class of input—one can decide whether a service curve is strict or non-strict.

By providing and proving, firstly, theorems to get weaker forms of strictness and, secondly, a more general approach to get service curves or worst case delay bounds, we bypassed this difficulty. Therefore, deterministic performance analysis based on NC in situations comprising aggregate scheduling remains applicable.

References

[1] U. Klehmet, T. Herpel, K.-S. Hielscher, and R. German, "Worst Case Analysis for Multiple Priorities in Bitwise Arbitration," in *GI/ITG-Workshop MMBnet 2007, Hamburg, pp. 27-35*.
[2] J. Liebeherr, A. Burchard, and F. Ciucu, "Delay Bounds in Communication Networks with Heavy-tailed and Self-similar Traffic," *IEEE Trans. Inform. Theory*, vol. 58(2), pp. 1010–1024, 2012.
[3] R. Cruz, "A calculus for network delay, part i: Network elements in isolation," *IEEE Trans. Inform. Theory*, vol. 37-1, pp. 114–131, 1991.
[4] J.-Y. Le Boudec and P. Thiran, *Network Calculus*. Springer Verlag LNCS 2050, 2012.
[5] Y. Ying, F. Guillemin, R. Mazumdar, and C. Rosenberg, "Buffer Overflow Asymptotics for Multiplexed Regulated Traffic," *Performance Evaluation*, vol. 65-8, 2008.
[6] A. Charny and J.-Y. Le Boudec, *Delay Bounds in a Network with Aggregate Scheduling*. Springer Verlag LNCS 1922, 2000.
[7] J. Schmitt, F. Zdarsky, and M. Fidler, "Delay Bounds under Arbitrary Multiplexing," *Technical Report*, vol. 360/07, 2007.
[8] T. Herpel, K.-S. Hielscher, U. Klehmet, and R. German, "Stochastic and Deterministic Performance Evaluation of Automotive CAN Communication," *Computer Networks*, vol. 53, pp. 1171–1185, 2009.
[9] S. Kerschbaum, K.-S. Hielscher, U. Klehmet, and R. German, "A Framework for Establishing Performance Guarantees in Industrial Automation Networks," in *Proceedings MMB and DFT 2014, Bamberg, pp. 177-191*, March 2014.
[10] M. Fidler and V. Sander, "A Parameter based Admission Control for Differentiated Services Networks," *Computer Networks*, vol. 44, pp. 463–479, 2004.
[11] G. Rizzo, "Stability and Bounds in Aggregate Scheduling Networks," Ph.D. dissertation, Ecole Polytechnique Federale De Lausanne, 2008.
[12] J. Schmitt, F. Zdarsky, and I. Martinovic, "Improving Performance Bounds in Feed-Forward Networks by Paying Multiplexing Only Once," in *Measurements, Modelling and Evaluation of Computer and Communication Systems(14th GI/ITG Conference), Dortmund*, March 2008.
[13] A. Bouillard, B. Gaujal, and S. Lagrange, "Optimal Routing for End-to-end Guarantees: the Price of Multiplexing," in *Valuetools '07, Nantes*, October 2007.

 ISBN: 978-1-61208-398-8